AD-A260 148

1989 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

JOHN F. KENNEDY SPACE CENTER
UNIVERSITY OF CENTRAL FLORIDA

INVESTIGATION OF IGES FOR CAD/CAE DATA TRANSFER

PREPARED BY:	Dr. George W. Zobrist
ACADEMIC RANK:	Professor
UNIVERSITY AND DEPARTMENT:	University of Missouri - Rolla Department of Computer Science
NASA/KSC	
DIVISION:	Data Systems
BRANCH:	Technical/Information Systems CAD/CAE
NASA COLLEAGUE:	Mr. Jerry Barnes
DATE:	July 28, 1989
CONTRACT NUMBER:	University of Central Florida NASA-NGT-60002 Supplement: 2

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
CR-166837		

4. Title and Subtitle	5. Report Date
1989 Research Reports NASA/ASEE Summer Faculty Fellowship Program Kennedy Space Center - University of Central Florida	October 1989
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Dr. E. Ramon Hosler, Editor Mr. Dennis W. Armstrong, Editor	
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
NASA - Kennedy Space Center Systems Training and Employee Development Branch Kennedy Space Center, Florida	NASA Grant NGT-60002 Supplement 2
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
NASA Headquarters University Programs Branch Washington, D.C.	Final Contractor Report 6/89 - 8/89
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This contractor's report contains all sixteen final reports prepared by the participants in the 1989 Summer Faculty Fellowship Program. Reports describe research projects on a number of different topics.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Human-computer interface software, metal corrosion, rocket triggered lightning, automatic drawing, 60-Hertz power, carotid-cardiac baroreflex, acoustic field, robotics, AI, CAD/CAE, cryogenics, titanium, flow measurement	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	418	

NASA FORM 1626 OCT 86

1989 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

JOHN F. KENNEDY SPACE CENTER
UNIVERSITY OF CENTRAL FLORIDA

INVESTIGATION OF IGES FOR CAD/CAE DATA TRANSFER

PREPARED BY:	Dr. George W. Zobrist
ACADEMIC RANK:	Professor
UNIVERSITY AND DEPARTMENT:	University of Missouri - Rolla Department of Computer Science
NASA/KSC	
DIVISION:	Data Systems
BRANCH:	Technical/Information Systems CAD/CAE
NASA COLLEAGUE:	Mr. Jerry Barnes
DATE:	July 28, 1989
CONTRACT NUMBER:	University of Central Florida NASA-NGT-60002 Supplement: 2

ACKNOWLEDGMENTS

This is to acknowledge the support of Mr. Jerry E. Barnes and Mr. Hank Perkins of DL-DSD-22, Mr. Bill Higgs of EG&G, Mr. Ed Scarbourough and Mr. Don Chappell of Lockheed Space Operations, and Mr. Frank Kapr, TV-TIO. They were very helpful in the initial guidance of this research effort and in obtaining the necessary resources. Mr. L. Purves and Mr. S. Gordon, NASA/Goddard, provided an IGES test file and helpful comments.

Additionally, I wish to acknowledge the administrative support of Dr. R. Hosler, University of Central Florida and Mr. Dennis Armstrong, Training Section, NASA/KSC.

ABSTRACT

In a CAD/CAE facility there is always the possibility that one may want to transfer the design graphics database from the native system to a non-native system. This may occur because of dissimilar systems within an organization or a new CAD/CAE system is to be purchased. The Initial Graphics Exchange Specification (IGES) was developed in an attempt to solve this scenario. IGES is a neutral database format into which the CAD/CAE native database format can be translated to and from. Translating the native design database format to IGES requires a pre-processor and translating from IGES to the native database format requires a post-processor.

IGES is an artifice to represent CAD/CAE product data in a neutral environment to allow interfacing applications, archive the database, interchange of product data between dissimilar CAD/CAE systems, and other applications.

The intent of this paper is to present test data on translating design product data from a CAD/CAE system to itself and to translate data initially prepared in IGES format to various native design formats. This information can be utilized in planning potential procurement and developing a design discipline within the CAD/CAE community.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

In a CAD/CAE facility there is always the possibility that one may want to transfer the design graphics database from the native system to a non-native system. This may occur because of dissimilar systems within an organization or a new CAD/CAE system is to be purchased. The Initial Graphics Exchange Specification (IGES) was developed in an attempt to solve this scenario. IGES is a neutral database format into which the CAD/CAE native database format can be translated to and from. Translating the native design database format to IGES requires a pre-processor and translating from IGES to the native database format requires a post-processor.

IGES was developed in 1979 under direction of the National Bureau of Standards and several industrial concerns. Version 1.0 of IGES was published as part of an ANSI standard in 1981, Version 2.0 in 1983, Version 3.0 in 1986, and Version 4.0 in 1988 (ref. 1,2,3,4).

Version 1.0 supported CAD/CAE geometries, annotation entities, wireframe entities and some surfaces, Version 2.0 additionally supported finite element modeling, printed circuit board models, more text fonts, and extended some of the geometrical entities, Version 3.0 added additional surfaces, clarification of view and drawing entities, enhanced MACRO capability, plant flow and ASCII compression, and Version 4.0 supports solid models, enhanced electrical and finite element applications, and introduction of architecture/engineering/construction applications.

IGES is an artifice to represent CAD/CAE product data in a neutral environment to allow interfacing applications, archive the database, interchange of product data between dissimilar CAD/CAE systems, and other applications.

Developers must write software to go from the native database format to the IGES neutral database, and vice versa, since IGES is a specification and not a product. Therefore the IGES file is only as good as the developer's effort in this regard. In general, IGES is a superset of a CAD/CAE systems entity menu.

The intent of this paper is to present test data on translating design product data from a CAD/CAE system to itself and to translate data initially prepared in IGES format to various native design formats. This information can be utilized in planning potential procurement and developing a design discipline within the CAD/CAE community.

II. USAGES OF NEUTRAL DATA FILE

The concept of the neutral data file was in usage before IGES was developed through the development of database interfaces by various vendors. These interfaces were normally used by application engineers to write programs of use to the design organizations. One example was the development of a Motor Control Center (MCC) placement and one-line diagram drawing by interfacing vendor catalog information, MCC module placement algorithms, and drawing commands through the host neutral data file (ref. 5).

This neutral datafile contained the drawing command structure to enable the application engineer to invoke various graphics design entities, such as lines, circles, points, text, etc.

This concept is useful as long as one is utilizing a single vendor for the applications and the system will not be changed in the forseeable future. Once the CAD/CAE system is changed then the application programs can not utilized since the graphics commands will not normally be recognized by a different vendor. To achieve an environment whereby the product design data and applications could become stable requires a standard product design data interface. This accomplishment is attempted by IGES.

The concept of the neutral datafile can be utilized in more scenarios than transferring product data between dissimilar systems. One example was illustrated in the preceding paragraphs.

Various uses of the neutral graphics database follows (ref. 6):

a. A means for transferring product graphics design data between dissimilar CAD/CAE systems. This in principle allows design data to be represented in a neutral file so that it can be translated to a future CAD/CAE systems native graphics database. Thereby design drawings need not be re-drawn each time a new system is purchased, or if one is required to transfer graphics design data to another system for integration of electrical/mechanical information, or for checking by a facility which has a non-compatible system, etc.

b. As mentioned earlier one can develop application programs that utilize the neutral database format. These applications are useful in the design/analysis mode and pre-preparation of various design commands.

c. It is also possible to edit CAD/CAE drawings from a terminal rather than at a design workstation. This reduces editing time and a possible reduction in cost, due to the cost differential of terminals versus workstations.

d. Possibly one of the more useful applications of the neutral file concept is to archive design drawings. If the design graphics is stored in the native graphics format, it is probable that in the future the product design database would not be compatible with the CAD/CAE system in usage at that time, even if it was from the same vendor. Once the graphics is in a neutral format, one can in principle write a post-processor to translate the neutral database to the present native design format. This translator can be utilized on all archived drawings that are to be installed on that particular system.

e. One can envision various artificial intelligence (AI) type applications utilizing an expert system that will operate upon the neutral database. Possible applications could be, rules that allow interference checking in electrical/mechanical/piping drawings, rules for printed circuit board physical layout, integrated diagnostics (ref. 7), etc. One could also envision development of an expert system that checked a drawing for completeness, i.e., a rectangle which is not closed, as a simple example. If the expert system is designed around the neutral file database, then if the native format changes this should not disturb the algorithms developed.

It should be noted that in practice most of these would be difficult to achieve with IGES in it's present form. This will be discussed in a later section.

III. GENERIC COMPONENTS OF PRODUCT DATABASE

The major components of a generic product database are the following (ref. 8):

3.1 FORMAT

Formatting refers to the various bit representations in a system, i.e., character, floating point, fixed point, and integer being the most common ones. This manifests itself in the basic accuracy of the drawing and the character set representation. There is an inherent problem in matching the accuracy of the model generated to the model being transferred to another CAD/CAE system.

3.2 REPRESENTATION

This refers to how the geometry of a part is represented. There are several different schemes for part representation. A part can be represented by **edge boundary** or, **wireframe**. This is where the part's extremes are represented by a collection of curves in space. Other representations are, **surface** and **hybrid edge-surface**. The surface representation is more precise, especially for points not on an edge boundary, and the hybrid edge-surface is a combination of the preceding representations.

The representation principally provides the collection of geometrical parameters that make up each data element. For example, the representation of a line is it's end-points versus an equation with initial and final points.

3.3 MEANING

The meaning conveys the design intent of the data elements. One may have four lines connected in a rectangular pattern. This could either represent four disjoint lines or could represent a plane. To convey meaning one needs the concept of associativity whereby the four lines can be associated together, or not. This is a subtle concept since many times the meaning can only be conveyed by the user, unless associativity attributes are given.

IV. IGES FILE STRUCTURE

The IGES file structure (ref. 4), illustrated in Figure 1, is composed of six sections and they must appear in the following order:

a. Start - this section provides a human-readable prologue to the file. There must be at least one start record.

b. Global - this section contains the information to be used by the pre- and post-processor to translate the file. A sampling of the items contained in this section are; parameter and record delimiters used, information about sending system, file name, data format information, model space scale, user intended resolution. Basically, this section provides a definition of the global conditions under which the model was generated.

c. Directory - there is a directory entry for each entity in the file. This entry is fixed in size and contains twenty fixed format fields. This section provides an index for the file and attribute information about each entity. Typical attributes would be, line font, view, level, transformation matrix used, line weight, color, and form number.

d. Data - this section contains parameter data associated with each entity. This section has a free format structure. Typically, items in this section enable the graphics system to place the entity in the drawing. Therefore, this section contains placement data, pointers to properties/attributes of the entity, and back-pointers to associativity instances. The Data and Directory section comprise the representation of the entity and are used together.

e. Terminate - this section contains only one line and is fixed format. This record is used to total up the number of entries in the previous sections.

The Directory/Data sections result in redundant data and forward/backward pointers. This results in voluminous file size and abortive results if pointers are omitted or corrupted.

The IGES structure is a fixed length record of up to 80 ASCII characters. This allows for universal file readability, but it also is quite cumbersome. Although, later versions have the option of a compressed ASCII and Binary format which can be utilized to reduce file size. The compressed ASCII and Binary formatting addresses the volume of data, but imposes a processing burden compared to ASCII.

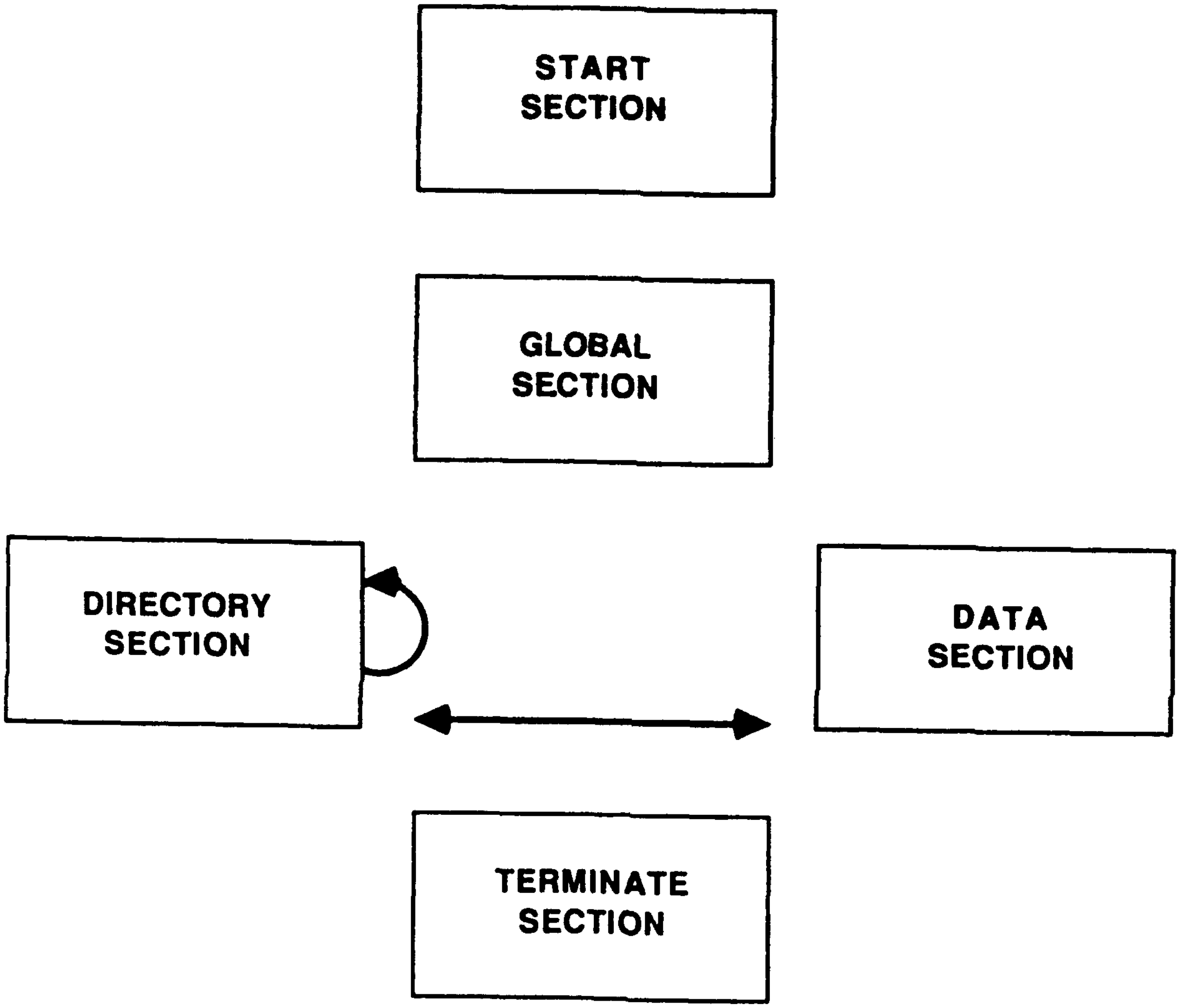

Figure 1 IGES File Structure Sections

V. IGES COMPONENTS OF PRODUCT DATA BASE

The format of the IGES file is 80 character ASCII records which detail the native system format. This also creates a large file structure.

There are four basic representations in IGES. They are, geometry, dimensioning and annotation, structure and properties. The main geometrical entities are the point, line, conic, arc, parametric spline, face, ruled surface, surface of revolution, and tabulated cylinder. These can be used to represent the basic graphical entities used in a drawing. Dimensioning and annotation are composed of text, arrowhead, and witness lines in various forms and styles. There are also special dimensioning entities such as, center lines. Splines are also represented, but different curves can result in the translation from a common set of input conditions. This is due to the host algorithm for representing a spline from it's input points and conditions. This is addressed in IGES by having a variety of spline forms.

IGES meaning is addressed through various structuring and property mechanisms. There are methods for assigning specific relationships between entities and also to convey meaning to these relationships. There are three important methods utilized. The associativity mechanism places specific entities into a group. An example, would be placing the four lines of a rectangle into a group representing a plane. Another mechanism is to place a group of entities into a view. This is a theoretical cube in which the entity group is placed and can be rotated and/or clipped. The final mechanism is the drawing which is a collection of view entities.

IGES also has the capability of the user defining properties.

VI. PROCESSING

The procedure for processing product design data is to translate from the native format to IGES by a pre-processor and a post-processor is utilized to translate from IGES to the native format. Many times this is done by developing pre- and post-processors to translate between the host's own neutral file. Intergraph translated IGES to/from the Standard Interchange Format (SIF) which is it's representation of the native graphics design database.

There are many difficulties associated with this task. There is not always a one-to-one mapping between the native graphics design format and IGES format. There can be one-to-many, many-to-one, and null translations. For example, the pre-processor must decide for a particular line font utilized by the host which IGES line font to use (one-to-many) and the post-processor must decide which native line font to use for a class of IGES line fonts (many-to-one). There is also the possibility that a particular native database entity has no IGES entity or vice-versa (null). For example, the native database may have only ellipse entities with the circle being a special case and IGES has both circles and elliptical entities, this will result in a null situation. There are also meaning conflicts; should a plane be represented as four connected line entities or a plane entity?

VII. IGES PROBLEMS

There are several problems which are typically encountered in utilizing the neutral database concept (ref. 9). They are; incomplete processors, poor choice of mapping, internal database organization has structural differences, and the user's choice of host drawing entities.

The incomplete processor problem must be addressed by the vendor since they are the one's who develop the translators between the native database format and IGES. Once this translator has been developed the user can not improve upon it. Although there may be some 'fine tuning' that could possibly be done through an expert system, if additional information could be obtained from the vendor on its native database structure.

The vendor has the responsibility for mapping choices. An example, would be whether a plane should be mapped into a separate entity, or mapped into it's constituent parts. Also, many times special symbols are preprocessed into a geometrical part, such as an ASCII character mapping into a particular arrowhead. Some of the mappings may be poor ones and hence difficult to recover through a re-translation.

Another problem is in how the host's internal data organization is represented. An example would be whether text should be free-standing or attached to the appropriate entity. The representation problem can result in unreadable drawings, caused by text overlapping, spacing problems, rotations, problems resulting from roundoff due to different numerical formats in vendor A and B. This is also, inherently, a result of how the vendor represents the model internally and little can be done by the user.

The last problem to be discussed is the user's choice of graphic entities. The entities that the user employs in the design process can result in efficient or inefficient translation of a drawing. If the user chooses and/or arranges entities that best suit the application and then when these are translated into IGES they may or may not be the best entities for re-translation to a design file. To address this problem the design organization can develop an IGES translation manual which lists host entities and their equivalent IGES entities, denoting if they are one-to-one, one-to-many, many-to-one, and null. This can result in user discipline in utilizing a set of host entities that are suited for translation. Of course, the problem is that user choice and innovation will be restricted.

VIII. OTHER NEUTRAL DESIGN FILES

Vendors have also developed their own neutral data files. Many of these formats are superior, in certain aspects, to IGES, but they are not industry standards and hence can normally only be utilized for the CAD/ CAE system for which they were developed. Typically they were designed as an application interface rather than for product data transfer. This section will briefly describe the format utilized by Intergraph and Autocad.

The Intergraph format is the Standard Interchange Format (SIF) and it has a relatively simple format, as shown in Figure 2. There are no forward and/or backward pointers, it is easily read and edited. It has only one entity record, as compared to the Directory/Data relationship in IGES. A disadvantage is that it is free format which requires a parser to read and interpret. Another disadvantage is that placement data is in UOR's rather than design units. A UOR is a drawing coordinate.

The Autocad format is called DXF and has a simple structure. Most of the file is fixed format and hence does not have to be completely parsed and interpreted. The format is simple enough, so that it can be edited from a terminal, as compared to IGES, although the files can be quite large. A disadvantage is that it doesn't support as many graphic entities as IGES.

One of the major advantages of IGES is that it accommodates most graphics entities that a design organization may require and does a reasonably good job with geometrical data. Disadvantages are; some translators are not fault tolerant, use forward/backward pointers in the Directory/Data section, errors in the pointer structure will destroy the entire drawing, difficult to edit, file transfer can be quite slow due to the large file size, e.g., a simple graphic line requires three entries in the Directory/Data section, see Figure 3, graphic entities may be transferred but their meaning lost, and at present very few translations are 100% correct.

The major difficulty with non-IGES neutral files is that they are not industry standards and typically not required by major industries and/or governmental agencies which utilize CAD/CAE services.

```
DID/NA=ZOBRIST2,DA=06/08/89,MO=3,RA=0,0,-1524000,4572000,0,1524000,DU=
    16909,1000,254,IN,ML,created by Intergraph SIF release 8.8 12-FEB-19
    88 REV2
LAC/LT=2
LAC/LC=3
BST/OP
LST/OP,0,0,762000,1524000,0,762000
ARC/CL,CE=2286000,0,762000,P1=1524000,0,762000,P2=3048000,0,762000,MA=
    1.,0.,0.,0.,0.,1.,0.,-1.,0.
LST/OP,3048000,0,762000,4572000,0,762000
EST/
BST/OP
LST/OP,0,0,-762000,1524000,0,-762000
ARC/CL,CE=2286000,0,-762000,P1=1524000,0,-762000,P2=3048000,0,-762000,
    MA=1.,0.,0.,0.,0.,-1.,0.,1.,0.
LST/OP,3048000,0,-762000,4572000,0,-762000
EST/
```

Figure 2 Intergraph Standard Interchange Format Structure

```
This IGES file was created by the INTERGRAPH IGES OUT translator        S      1
1H,,1H;,21HINTERGRAPH 8.8.5 IGES,12HZOBRIST2.IGS,10H8.8.5 IGES,         G      1
3H1.0,32,08,24,08,56,,1.0,1,4HINCH,32,,13H890608.144640,                G      2
 0.000003937007932,,17HSTANDARD PRODUCTS,10HINTERGRAPH,;                G      3
     124       1       0       0       0       0       0       0 0 0 0 0D      1
     124       0       0       1       0               MATRIX         1D      2
     110       2       1              40       0       0       00010100D      3
     110       2       3       1                       LINE           1D      4
     124       3       1       1       0       0       0       00000000D      5
     124       0       0       2                       MATRIX         0D      6
     100       5       1              40       0       5       00010000D      7
     100       2       3       2                       ARC            1D      8
     110       7       1              40       0       0       00010100D      9
     110       2       3       1                       LINE           2D     10
     102       8       1              40       0       0              D     11
     102               0       1                       COMPCURV       1D     12
     110       9       1              40       0       0       00010100D     13
     110       2       0       1                       LINE           3D     14
     124      10       1       1       0       0       0       00000000D     15
     124       0       0       2                       MATRIX         0D     16
     100      12       1              40       0      15       00010000D     17
     100       2       0       2                       ARC            2D     18
     110      14       1              40       0       0       00010100D     19
     110       2       0       1                       LINE           4D     20
     102      15       1              40       0       0              D     21
     102               0       1                       COMPCURV       2D     22
124,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0;                   1P      1
110,0.000000,0.000000,3.000000,6.000000,0.000000,3.000000;             3P      2
124,1.000000,0.000000,0.000000,9.000000,0.000000,0.000000,             5P      3
-1.000000,0.000000,0.000000,1.000000,0.000000,3.000000,0;              5P      4
100,0.000000,0.000000,0.000000,3.000000,0.000000,-3.000000,            7P      5
0.000000;                                                              7P      6
110,12.000000,0.000000,3.000000,18.000000,0.000000,3.000000;           9P      7
102,3,3,7,9,0;                                                        11P      8
110,0.000000,0.000000,-3.000000,6.000000,0.000000,-3.000000;          13P      9
124,1.000000,0.000000,0.000000,9.000000,0.000000,0.000000,            15P     10
1.000000,0.000000,0.000000,-1.000000,0.000000,-3.000000,0;            15P     11
100,0.000000,0.000000,0.000000,3.000000,0.000000,-3.000000,           17P     12
0.000000;                                                             17P     13
110,12.000000,0.000000,-3.000000,18.000000,0.000000,-3.000000;        19P     14
102,3,13,17,19,0;                                                     21P     15
S      1G      3D     22P     15                                        T      1
```

Figure 3 IGES File Structure.

IX. ALTERNATIVES

There are several alternatives to using IGES to transfer graphics data between dissimilar CAD/CAE systems.

One approach is to write a direct translator, i.e., one which translates the vendor A database directly to vendor B database. These translators usually are very efficient, since they address a particular problem. To build one of these translators requires knowledge about the data structure for each system for which there is to be a translator built. One possible technique is to utilize the vendors neutral file rather than IGES, such as, SIF or DXF.

Of course, if the CAD/CAE systems for which the direct translator is built is changed a new translator must be designed. This would require n(n-1) translators to be built, if graphics data is to be transferred between n dissimilar CAD/CAE systems, as illustrated in Figure 4. IGES only requires 2n pre/post-processors to be built for the same number of dissimilar systems, which is shown in Figure 5.

If a vendor changes the native database structure, then n-1 direct translators would have to be re-built, but only 2 pre/post-processors.

A new neutral file structure is being developed it is called Product Data Exchange Specification (PDES) (ref. 10). PDES is planned for release in the 1990's and defines a more conceptual model than IGES.

The model consists of an application layer, conceptual layer, and a physical layer. The application layer is concerned with the application, i.e., electrical, mechanical, architectural, etc. The conceptual layer is concerned with concepts such as, tolerance envelopes, solids with flanges, etc. and the physical layer is concerned with the manufacturing process, cost's, suppliers, numerical control tool paths, and layout drawings to mention a few.

If PDES is to replace IGES in the future it would have to be compatible, so that IGES files could be translated to PDES.

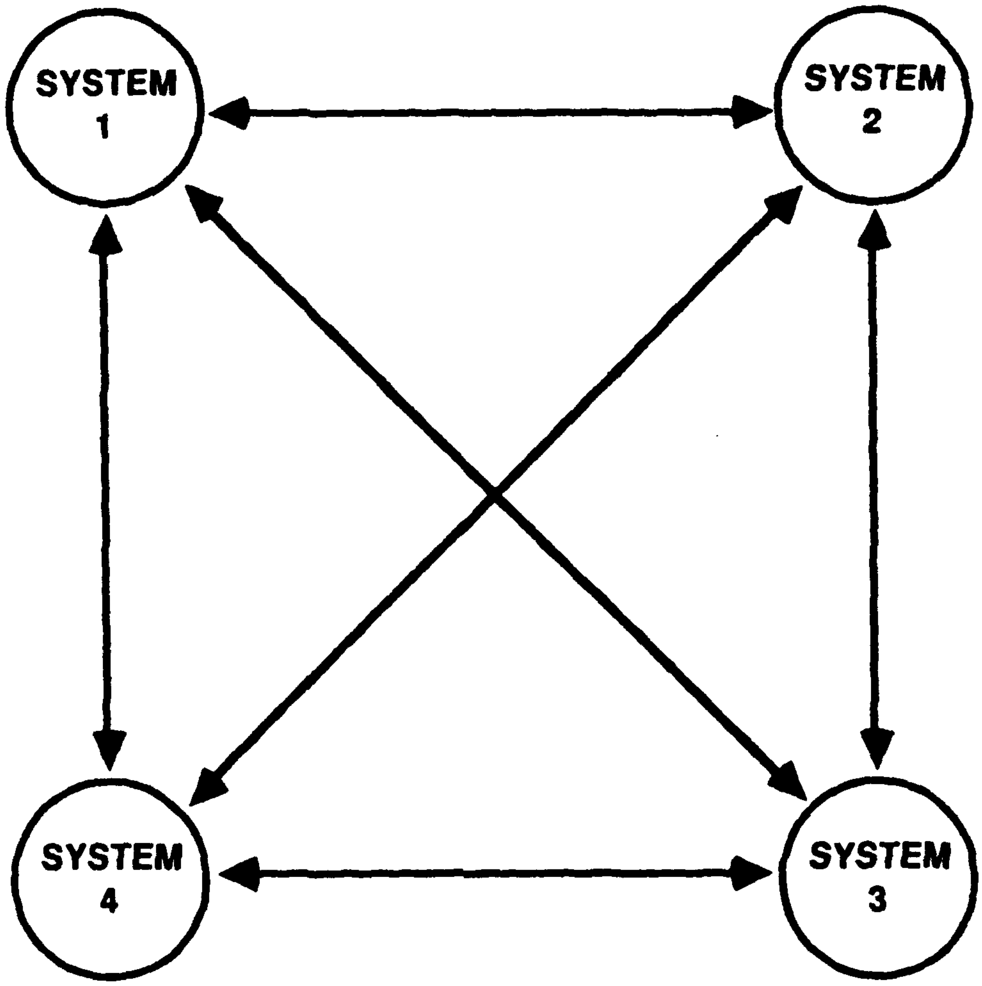

Figure 4 Direct Translation Process

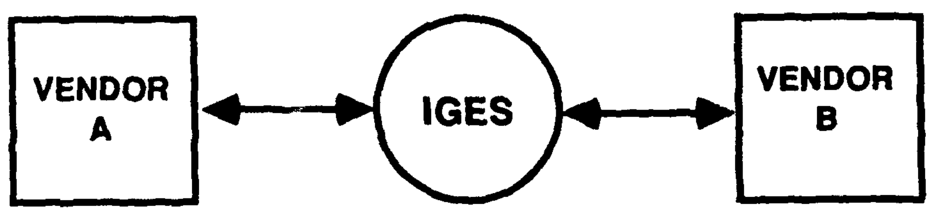

Figure 5 IGES Translation Process

X. TEST PROCEDURES

To evaluate an IGES translator one must perform several tests.

The IGES test files that are needed, are the following:

a. A test file that contains simple entities, mainly geometric to evaluate the basic translation process. These would be lines, points, circles, arcs, splines, etc and would provide a baseline.

b. Develop a test file with various entities, each enclosed in a box or separated. This would provide useful information on which entities transfer and also which native entity results.

c. A test file(s) that is a typical production part or schematic of a useful layout. This test file would be complex and give an indication of how reliable the translation process will be in the production environment. These file(s) should also include, if possible, a complex system that will be typical in the future.

There are various ways to evaluate the test results. One could compare plots through an overlay process, or a count of entities and their positions. This would give information as to how reliable the data is transferred and if in the same position. It is also important to see if the data elements can be manipulated. One could scale views, move geometric objects, place cells, edit text strips or dimensions, test to see if graphic groups are still graphic groups, etc.

More complex tests would be to test accuracy of curves, surfaces, and volume dimensions and postioning. This could be done for curves by creating a series of parallel lines through the curve and compare intersection points before and after translation. The same process could be used for surfaces and volumes by, respectively, using parallel planes and intersecting solids. These tests would be imperative if the drawing is used for analysis or direct measurements.

Any drawing that is translated will have to be verified that it corresponds to the original and validated, in the sense, that all functions will have to have been translated. This is no small task and has not been addressed thoroughly in this paper.

The quality of translation will most likely follow, in order of good to bad, the three tests outlined above.

XI. TEST RESULTS

Test translations were done with several CAD/CAE drawing packages with mixed results. The tests were performed with different levels of support and hence difficult to compare. Initially, simple geometrical parts were developed on the Intergraph CAD/CAE system and these were tested via a self-loop with success. Then a more complex part developed by an IGES test committee (ref. 11) was translated; as can be seen from Figure 6 and 7, the arrowheads and some attached text was lost, or mis-interpreted. Another self-loop test performed on the Intergraph system is shown in Figure 8. This test part was composed of lines and an arc mirrored. As can be seen from Figure 8, line fonts were mis-interpreted and a line was drawn through the arc endpoints. Figure 9 was a demonstration of how graphic group and cell entities were translated. In this case the graphic group was translated correctly but the cell capability was not translated. This was verified by bringing the design drawing up on the screen and then determining through menu commands if the circle and text was a graphic group or a cell.

The next suite of tests were for a drawing which contained 28 IGES entities, see Figure 10, and the Space Station. These IGES files were developed by NASA/Goddard, see ref. 12. The 28 entity file was translated by Intergraph (IGES version 8.8.5), AutoTrol series 7000 (on an Apollo platform), and the IBM CADAM package. The results of the 28 entity file are shown in Figures 11 and 12, for Intergraph and AutoTrol, respectively. The IBM CADAM system was unsuccessful in having the IGES file translated. The translation by Intergraph, see Figure 11, resulted in only one view, zero height text, and improper scaling. It should be noted that this was only accomplished after removing the B-spline entity from the design drawing, otherwise it killed the process. The translation by AutoTrol, see Figure 12, resulted in the four views being evident, but with some vector splash and certain entities missing, the main ones being surfaces of revolution. The AutoTrol drawings were translated with the help of an AutoTrol representative, while the Intergraph attempt was done by a design engineer. The 28 entity IGES file could not be translated by the IBM CADAM system.

The last IGES file translation attempted was for a very complex drawing. This is a drawing of the Space Station and the results of the translation by AutoTrol is shown in Figure 13. This translation is complete, since no translation errors were reported in the AutoTrol log. The translation by Intergraph resulted in only the border being displayed, and the IBM CADAM system was unable to translate. The translation of an IGES file containing solids entities was not attempted since the various CAD/CAE drawing packages either did not support solids, or could not translate the file (AutoTrol).

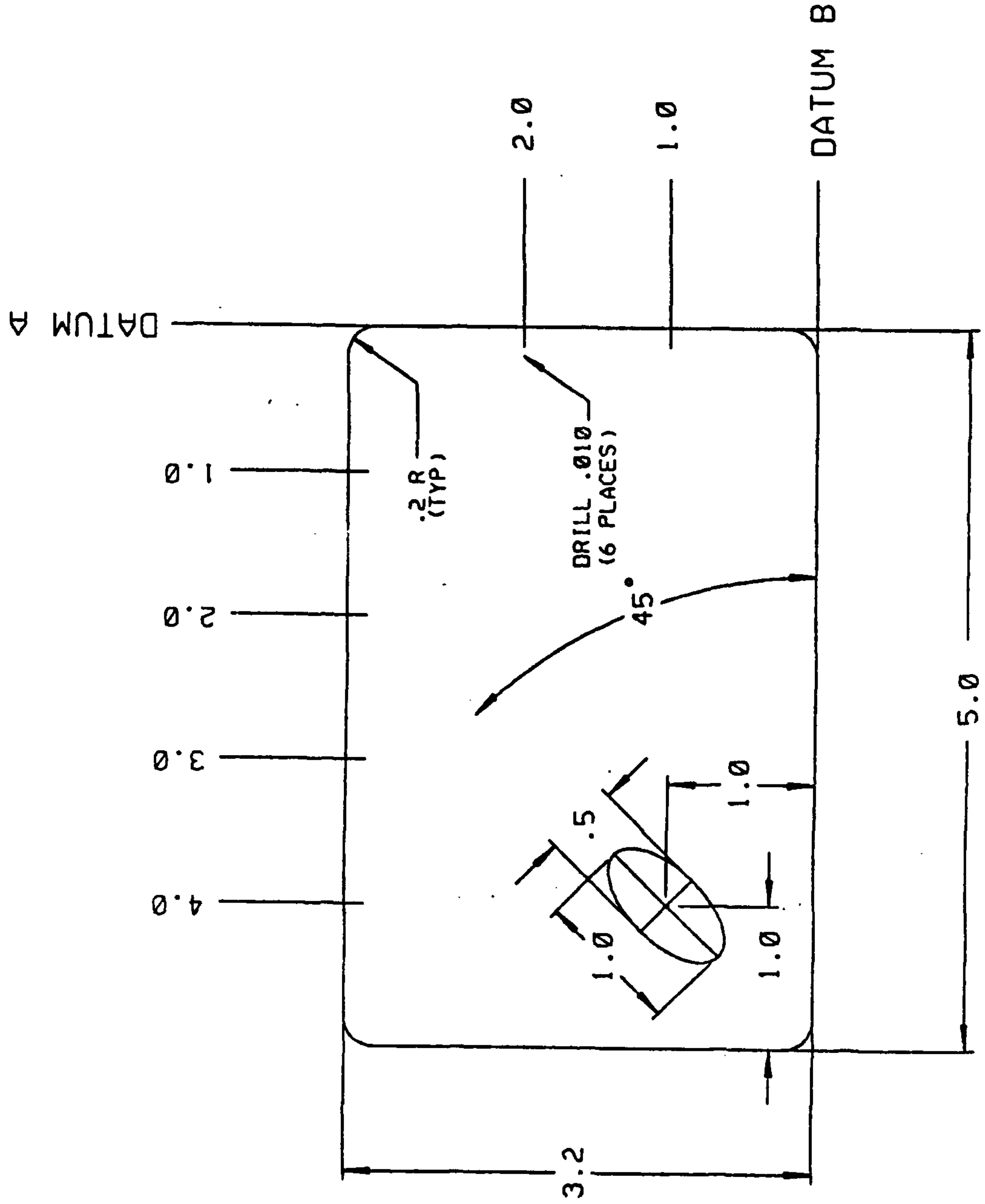

Figure 6 IGES Test Committee Part Geometry

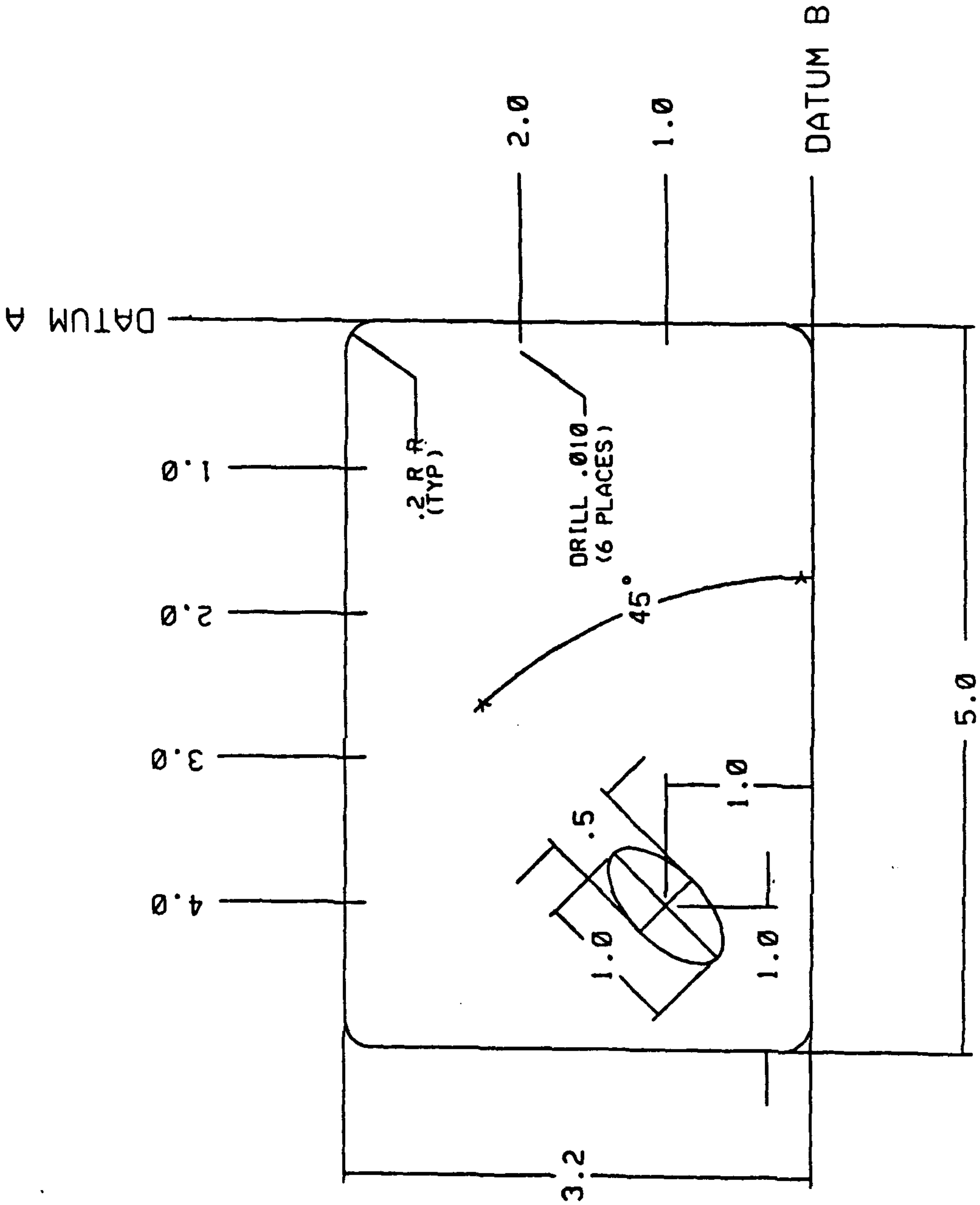

Figure 7   Translated Test Committee Part

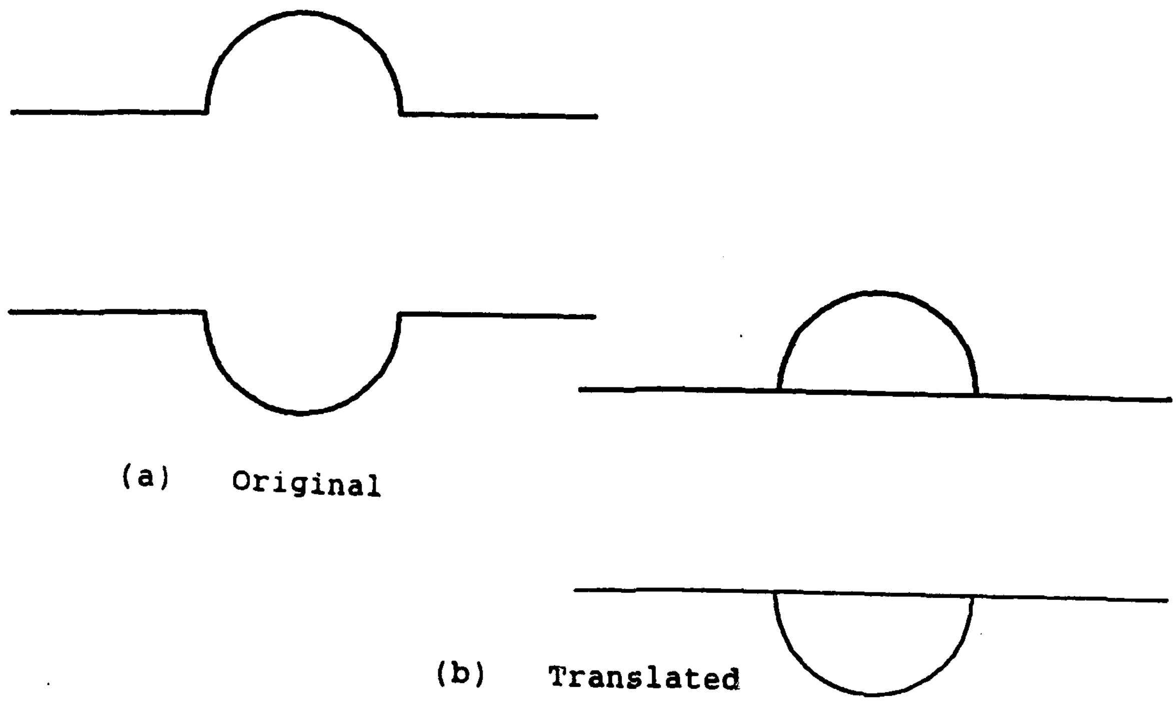

Figure 8 Mirrored Lines/Arcs Geometry

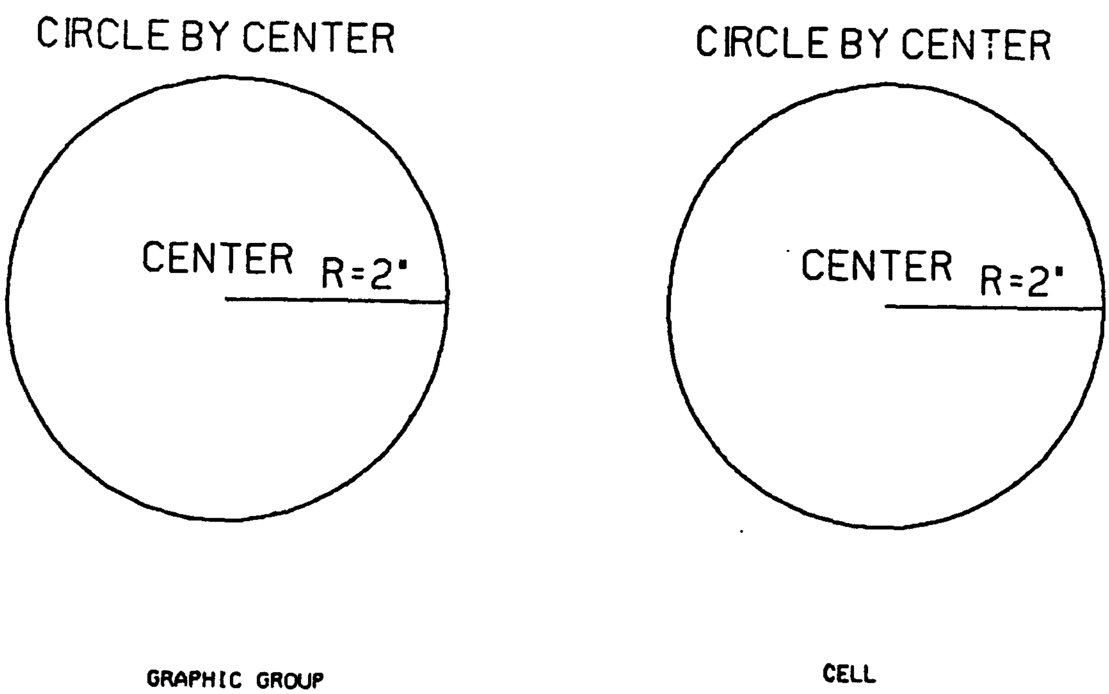

Figure 9 Translated Graphic Group/Cell Entities

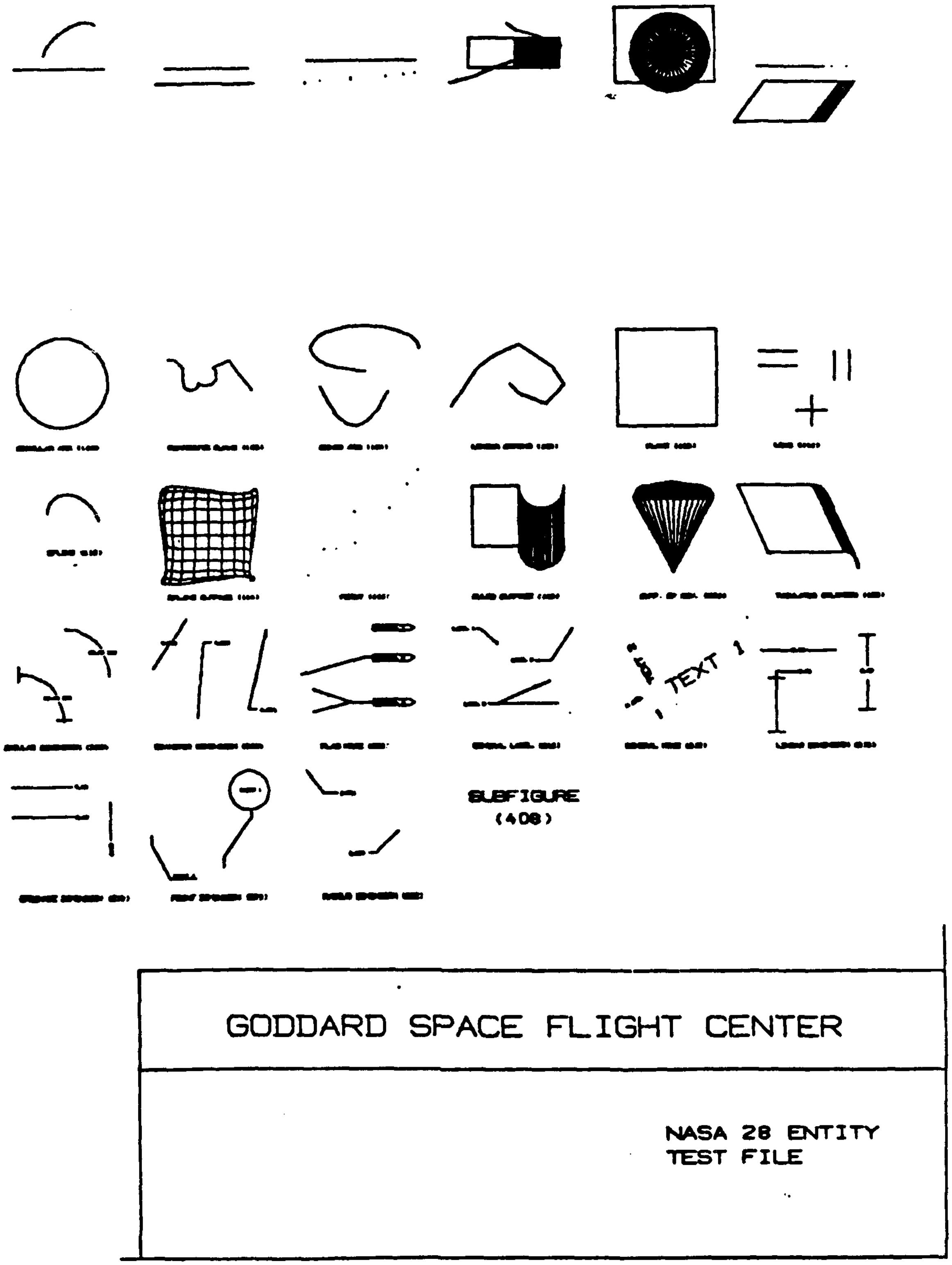

Figure 10 NASA/Goddard Four View 28 Entity File

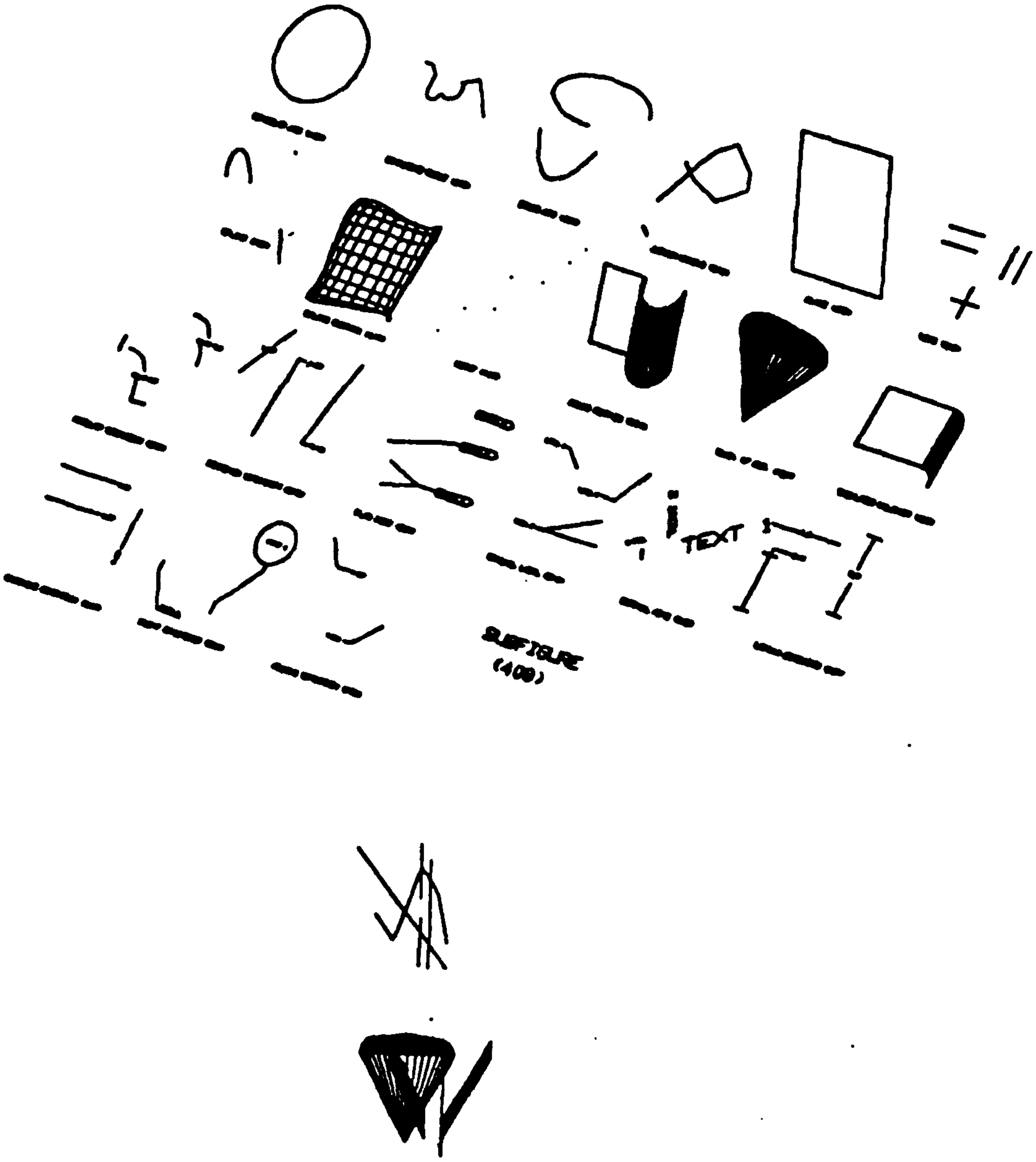

Figure 10 NASA/Goddard Four View 28 Entity File (Continued)

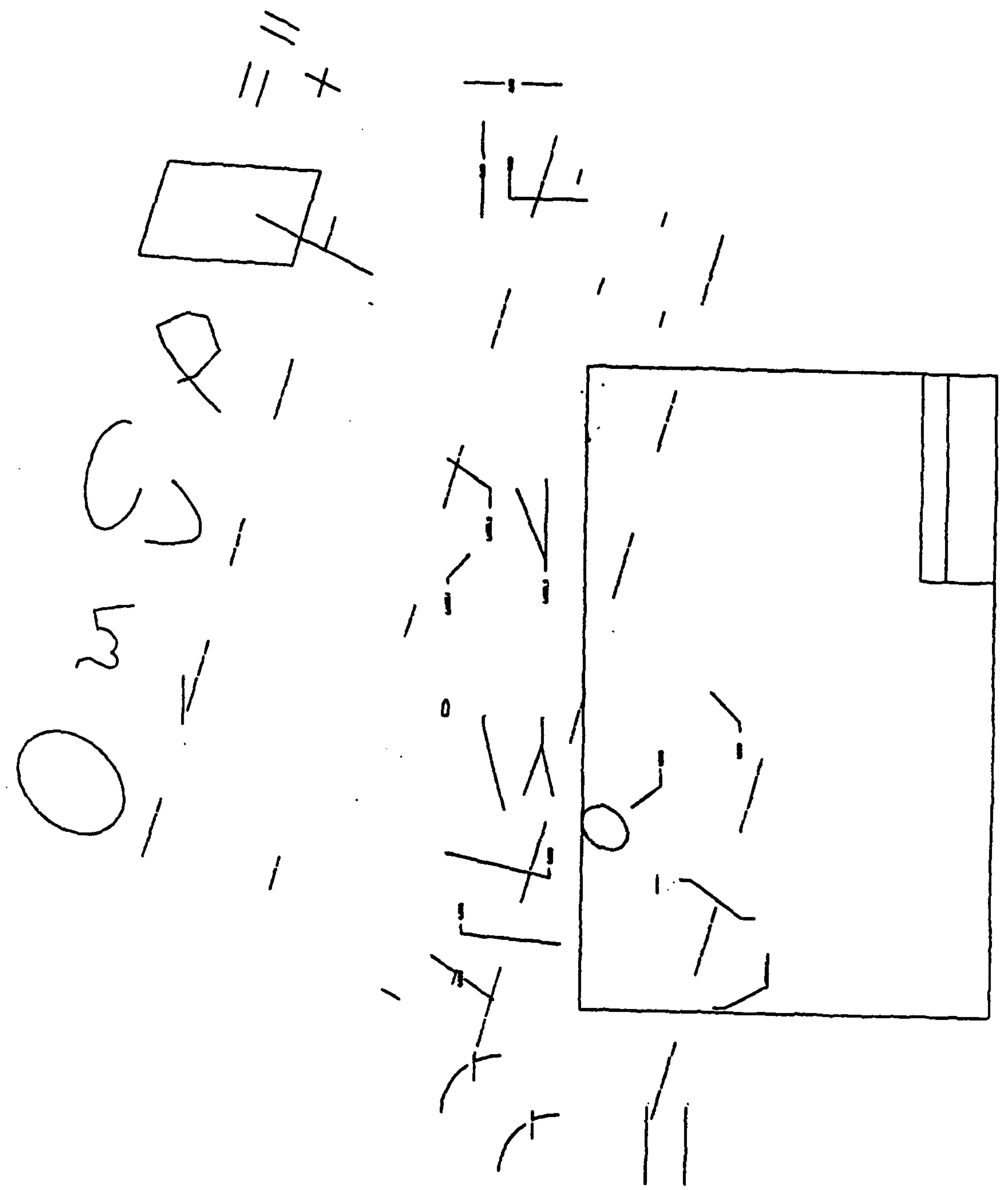

Figure 11 Translated NASA/Goddard 28 Entity File-Intergraph

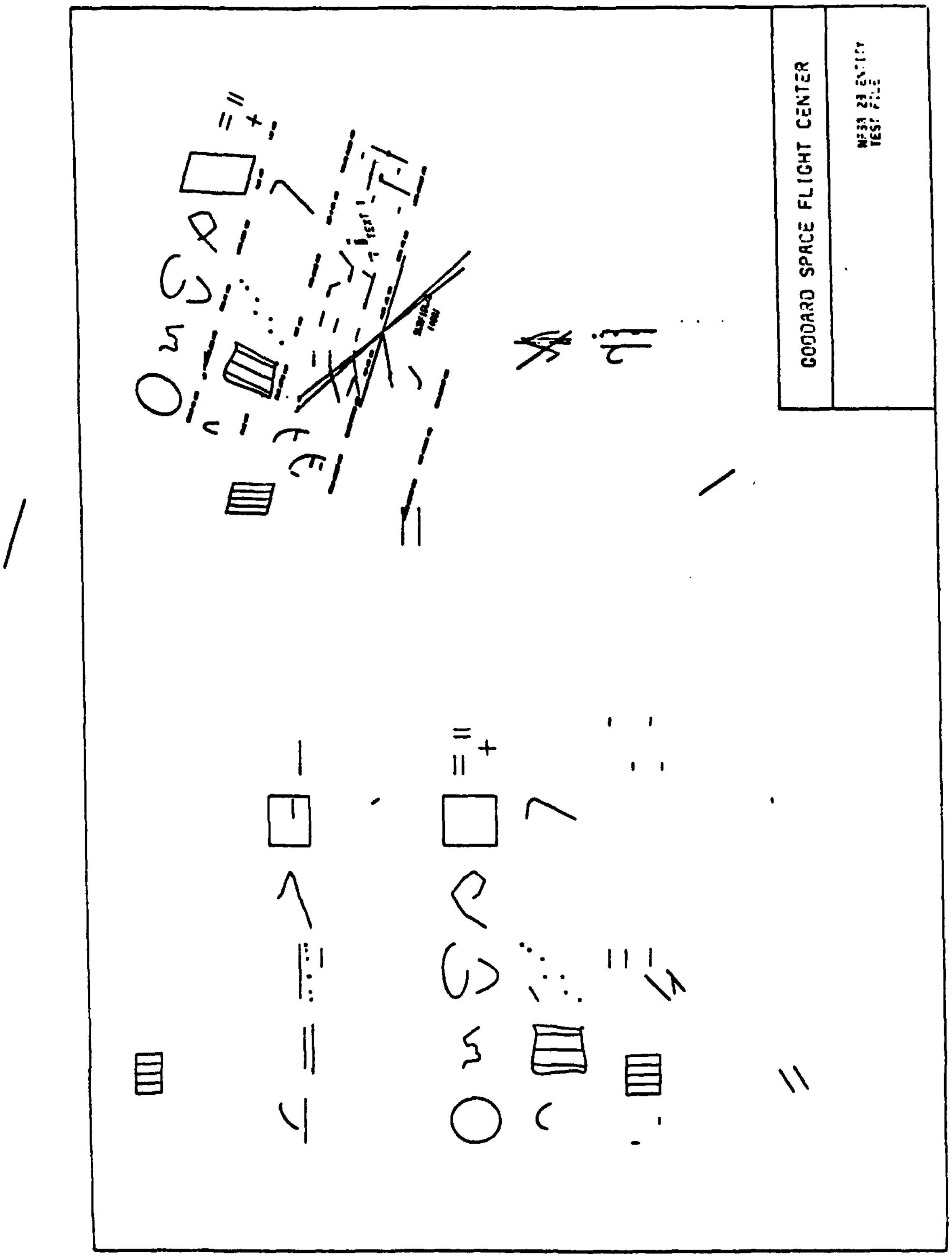

Figure 12 Translated NASA/Goddard 28 Entity File-AutoTrol

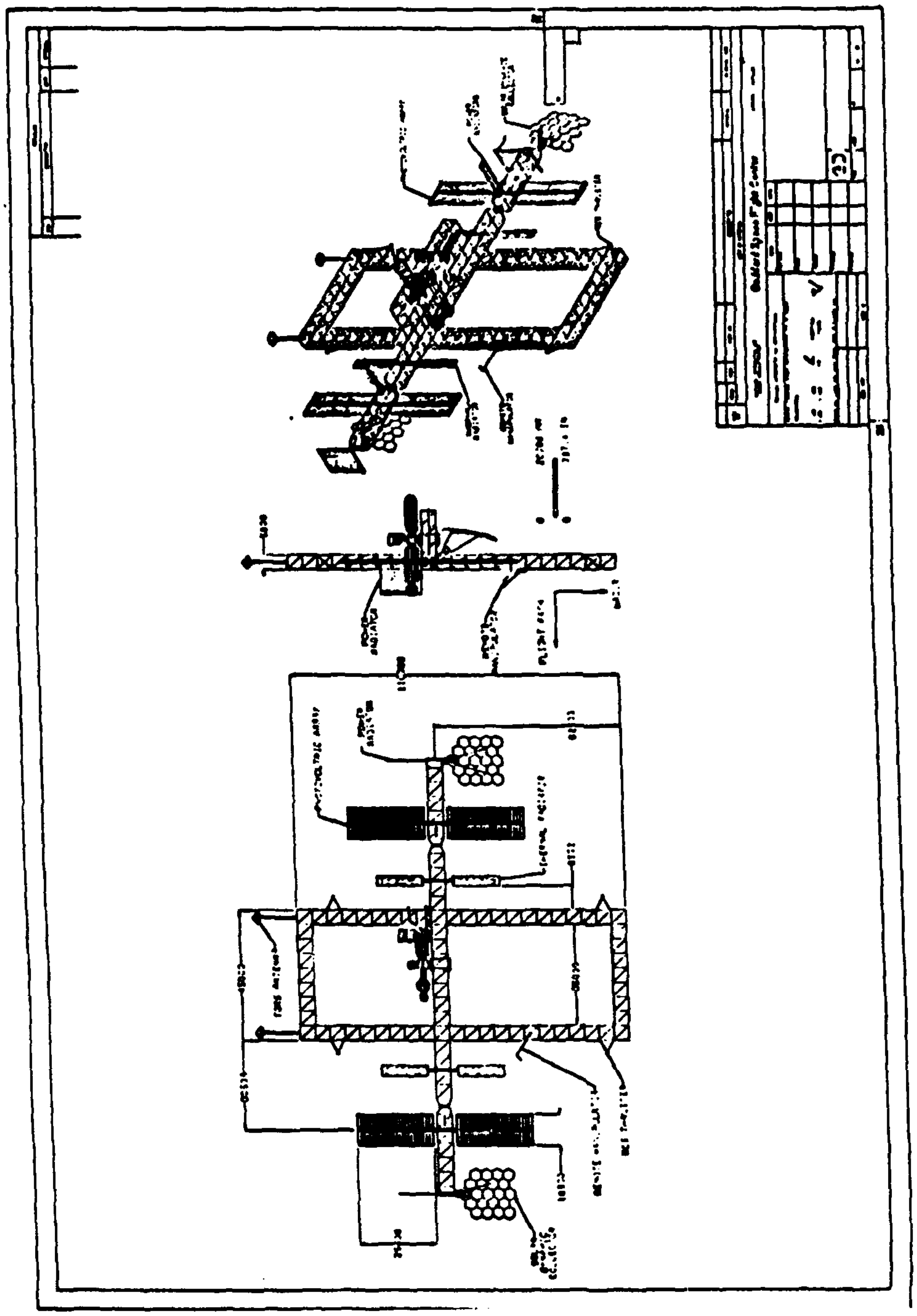

Figure 13   Translated NASA/Goddard Space Station-AutoTrol

XII. POSSIBLE STRATEGIES

There are several strategies that can be implemented to increase the success rate of IGES translations.

One area is to develop user discipline in the design environment. The designer needs to understand the relationship between the product the end user perceives and the set of computing elements that represent that product. They should be disciplined to utilize neutral database standards. This is probably easier said then done, since by doing this one will restrict the user's inovativeness, efficiency, and interest. This would involve the development of an engineering IGES handbook (ref. 13). This handbook would include a primitive set of entities that could be used in a particular engineering discipline, restricted forms within that entity that should be used, and lists of native to IGES entity translations.

This disciplined approach could be rigidly enforced in certain applications, i.e., those that will require possible translation, now or in the future. These are files that must be maintained for many years and when re-used would probably be modified or appended.

One approach to this is to build an auxiliary procedure involving table look-up which will translate user commands into acceptable IGES entities. This would not normally be done on-line, but only if a translation is to be done. This approach has been taken by Sandia Laboratories in the concept they call vanilla deflavoring and reflavoring (ref. 9).

These flavor translators convert IGES data acceptable to the sending system into IGES suitable for the receiving system. This is a better approach than using ad hoc procedures for editing the drawing when translating from vendor A to vendor B. It also eliminates hand-editing the IGES file, although this would be very difficult due to the complexity of the file structure. One still has the problem that the flavor translator is only as good as the pre/post-processing done by the vendor.

An example of the flavoring concept is in the conversion of line fonts to system line fonts, or deflavoring. These can then be re-translated to the closest line font at the other end, reflavor. Another example would be to decompose a composite into it's component parts, if the other vendor does not support.

Another approach would be to employ the **bubble-up** technique. This would require one to translate a design file only when needed and then verify/validate file and re-do entities as needed to make it a workable drawing. Possibly, only re-working those portions that are needed. This approach is probably valid if one is moving from vendor A to vendor B and a large number of drawings are currently residing on vendor A. In conjunction with this, it might be acceptable to only scan in drawings and then modify the scanned drawing, as needed, at a workstation.

For the initial translation a viable alternative would be a one-to-one translator, especially, if it is a one time transfer and not one that is continually occurring between numerous dissimilar systems.

The most important strategy, if one is to purchase a system that is from a vendor different than the one presently available and if there are design files to be transferred to the new vedors system from the existing vendor, is to require that the vendor must successfully perform the tests in Section X. Only by requirements such as these will the vendors put more effort into developing efficient IGES translators. Although, it should be noted that translating files from an existing vendor's graphic design database is only as good as the available pre-processor.

Finally, one could absorb the cost of translation when going from vendor A to B. Until more vendors have efficient IGES translators one is probably doing this anyway through development cost pass-through, but as more procurements require efficient translators to be built this development cost should become less.

As a final note the Engineering Design organization should consider dedicating a person(s) to keeping up with and understanding the nuances of the translation process.

XIII. SUMMARY AND CONCLUSIONS

The translation process is a difficult one and should be planned for in any procurement process and on-going design environment.

One should view the IGES translation, or any automated translation process, as the first step in obtaining a viable design drawing. Probably, in practice one should be able to obtain 70 - 90% of the drawing transferred correctly. This assumes that the vendor has developed an efficient pre/post processor. If the vendor has not developed and maintained an efficient set of processor's there is little the user can do to enhance the translation process.

The experience gained from obtaining translated drawings for the different test classes follows what one might expect, i.e., the more complex the drawings are - the more difficult to translate, the more experienced technical resources that are available - the more successful the translation, and certain vendors have better pre/post processors than others.

The solution to the translation process is not easily solved since there are conflicting goals. The engineering design organization would like to have a homogeneous architecture, but this is impractical due to the following reasons; responsibility is normally distributed in a large design organization and competition among vendors results in enhanced products that are very attractive to the user. Therefore, one can assume that the design environment will be heterogeneous.

In conclusion, the design organization should make test translations part of the procurement, user's should be aware of IGES capabilities, design standards should incorporate IGES capabilities when drawings are to be maintained for many years or modified, and there should be a dedicated group (or, person(s)) involved in IGES translations and their nuances.

A final reminder, remember that an IGES translation environment is only as good as the pre/post processors developed by the vendor.

XIV. REFERENCES

1. Nagel, R. N., Braithwaite, P. R., and Kennicott, P. R., Initial Graphics Exchange Specification (IGES), Version 1.0, NBSIR 80-1978 (R), National Bureau of Standards, 1980.

2. Initial Graphics Exchange Specification (IGES), Version 2.0, NBSIR 82-2631 (AF), National Bureau of Standards, February 1983.

3. Smith, B. and Wellington, J., Initial Graphics Exchange Specification (IGES), Version 3.0, NBSIR 86-3359, National Bureau of Standards, April 1986.

4. Smith, B., Rinauldot, G. A., Reed, K.A., and Wright, T., Initial Graphics Exchange Specification (IGES), Version 4.0, NBSIR 88-3813, National Bureau of Standards, June 1988.

5. Zobrist, G. W., "CAD Program for Automated Motor Control Center Layout", Proc. of Autofact West-CAD/CAM VIII, Los Angeles, CA, November 1980.

6. Keller, C., "IGES gets an F", MICROCAD News, September / October 1988, PP. 60 - 65.

7. Beazley, W. G., "Expert System Rules from CAD Databases", Proc. Air Force Workshop on Artificial Intelligence for Integrated Diagnostics, University of Colorado, Boulder, CO, July 29 - 31, 1986, PP. 292 - 297.

8. Liewald, M. H. and Kennicott, P. R., "Intersystem Data Transfer via IGES", IEEE Computer Graphics and Applications, Vol. 2, May 1982, PP. 55 - 58.

9. Ames, A. L. and Fletcher, S. K., " Making IGES Work for CAD/CAM Data Exchange", Proc. Autofact 1985, Detroit, MI, November 4, 1985.

10. Kelly, J. C., "The Product Data Exchange Standard (PDES), Federal Computer Conference, September 9, 1985.

11. Smith, B. M. and Wellington, J., "IGES, A Key Interface for CAD/CAM System Integration", CAD/CAM Technology, Vol. 3, Spring 1984.

12. Purves, L. R., NEXUS/NASCAD - NASA Engineering Extendible Unified Software System with NASA Computer Aided Design, NASA Goddard Space Flight Center, MD, M86-1209, 1986.

13. Hammond, D. and Wesenberg, R., "Progress Report: Development of an IGES Handbook for KSC", Unpublished Technical Summary, Engineering Development Directorate, NASA/KSC, FL, September 1986.